Global Information Assurance Certification Paper

SANS GIAC Security Essentials Certification - GSEC
Practical Assignment
Version 1.4c - Option 2
21 September 2004

Securing Webmin with Tcp Wrappers and SSH Port Forwarding – A Practical and Economical Approach

By: Howard Dulberg

Table of Contents

Abstract

More and more companies are utilizing software developed and maintained by the Open Source community to handle a multitude of security and administrative functions. Economics -- relating to software and/or labor costs -- play a critical role in the decision process, and companies are looking to replace expensive applications, which in most cases run only on the original platform for which they were purchased. It also costs a lot of money to upgrade and renew licenses on an annual basis. Labor cost can also play a role in how a company justifies application to be used. With the old convention of applications purchased from a vendor, you need someone who has been trained on both the hardware and software to ensure immediate productivity, or you must invest time and money training employees. This becomes an ongoing investment as attrition and turnover occur. Open Source software minimizes costs considerably. There is no cost for the software itself; open source is typically easy to use and train; and the software can be loaded on an employee's personal computer and learned at home. The advent of Open Source developed on Intel platforms allows for portability and scalability to other systems. A user can readily download the software on their PC and quickly get up to speed. Once the package has been learned, generally you can scale to other platforms with relative ease.

Once a decision is made to utilize Open Source freeware such as Webmin and after a carefully planned assessment and certification period -- which includes properly securing the application and ensuring proper TCP and SSH configurations -- Webmin can be used to handle all user account administration functions. Webmin comes with many more options including an administrator function that allows you to easily build Webmin groups to segregate groups of people or teams, allowing you to control what a user of Webmin can do. This includes setup and deletion of user accounts, and resetting user passwords. Webmin is a tool that allows the often cryptic acts of administrating a server to be done much easier, allowing lower-level support personnel to handle user account administration and/or manage different Webmin application clusters and functions. This frees up more experienced administrators to do higher-level work. Therefore, the purpose of this paper is to detail the steps required to download Webmin and ensure that the product is safe from hackers both inside and outside the company. This paper will detail the benefits of securing the Webmin product by using tcp wrapper and utilizing ssh port forwarding. Without these safeguards in place, the port utilized by the Webmin install is vulnerable to attacks and a nuisance during Nessus scans. This paper will specifically address securing the software and not the operational advantages or disadvantages of Webmin.

Section 1: Before Implementing the Software

Guidelines must be set to clearly define levels of responsibility and authority over the Webmin product and the use on any clients or servers attached to your network. Clear segregation of duties need to be established, and care taken to ensure confidentiality, integrity, and availability of both the application and the data. This should be done as part of an initial screening of any Open Source or purchased products and decided upon with the teams involved. In this case, the decision was to minimize what any group or team would be able to access.

Another good idea is to involve as many business owners of the particular servers and their applications as possible to ensure their comfort level with the user account administration process. In today's business market you may find that portions your IT staff can be paid or have a dotted line reporting structure to business owners. This situation plays a crucial role in the economic and labor costs we have already discussed. Without first achieving buy-in, you leave yourself open to having frustrated business owners, users and administrators.

It is also good to discuss with the teams how you can save time and money by implementing a secure product. This includes how securing your outward facing ports to the world can leave you open for hackers or intruders. In my experience, business owners and users tend to think that security measures are not necessary or are a nuisance. This is generally due to their limited knowledge and understanding of what a secure system can protect them from, as maybe they have not yet been affected in a negative way (i.e worms, hackers, etc). Securing your server ports and running scans can provide numerous benefits and safety.

Since we have decided to secure Webmin with tcp wrappers, it is important to understand why tcp wrappers were chosen instead of using a firewall. TCP wrapping is done at the port level; if you change the port where you originally installed Webmin, you will only need two changes. One change is to the port mapping entry in /etc/services. The other change is needed in your ssh port forwarding entry on the client. With a firewall, you need to change rules and port entries, which can be cumbersome compared to the tcp wrapping option. Although most packages will rarely change after the initial installation, if this does occur, you will be happier that you used tcp wrappers. When configured properly, both tcp wrappers and firewalls essentially accomplish this same level of security.

Section 2: Server Configuration Considerations and Examples for OpenSource ssh.

This section will delve into the server settings for Open SSH software, which addresses AIX server file configuration, and their AIX relationships to settings and files.

It is critical to understand how to properly configure ssh on the AIX server to allow for connection and port forwarding. Improper configuration can and will impede the overall hardening and securing of any service or ports. In this example of securing Webmin on an AIX server, careful consideration needs to be planned and executed.

Anyone can download OpenSSH clients (See section 3 below) on their PC's or laptops. To connect to an OpenSSH server from a client machine, you must have the OpenSSH clients and OpenSSH packages installed on the client machine.
I will be defining the configuration for an OpenSSH software tool, running on RedHat 9.0 supporting the SSH1 and SSH2 protocols. The importance of these tools and the main reason I used them was to ensure that network traffic is encrypted and authenticated. OpenSSH runs the sshd daemon process on the AIX host and waits for the connection from the clients. Although OpenSSH supports public and private-keys for authentication and encryption, I have not used these keys for the purpose of installing Webmin.

This section explains step-by-step, how to configure OpenSSH. Before installing OpenSSH format packages, you must install the Open Secure Sockets Layer (OpenSSL) software. Since I installed Webmin on a platform already configured, OpenSSL is not being covered in the paper.

The following general information covers OpenSSH:

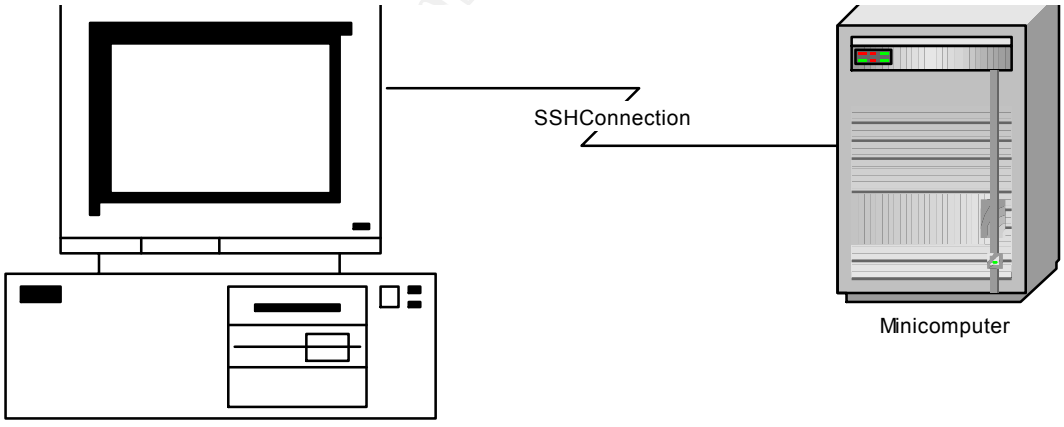
- The /etc/ssh directory contains the sshd daemon and the configuration files for the ssh client command.
- The /usr/OpenSSH directory contains the readme file and the original OpenSSH open-source license text file. This directory also contains the ssh protocol and Kerberos license text.
- When the OpenSSH server side files have been installed, an entry is added to the /etc/rc.d/rc2d directory.
- The sshd daemon automatically starts at boot time. This is because of an entry in inittab. The entry in inittab is to execute run-level 2 processes (l2:2:wait:/etc/rc.d/rc 2). To prevent the daemon from starting at boot time, remove the /etc/rc.d/rc2.d/Ksshd and /etc/rc.d/rc2.d/Ssshd files.
    - 0 - halt (Do NOT set initdefault to this)
    - # 1 - Single user mode
    - # 2 - Multiuser, without NFS (The same as 3, if you do not have

networking)
    - # 3 - Full multiuser mode
    - # 4 - unused
    - # 5 - X11
    - # 6 - reboot (Do NOT set initdefault to this)
- OpenSSH software also logs information to SYSLOG.

The IBM Redbook, *Managing AIX Server Farms*, provides information about configuring OpenSSH in AIX and is available at the following Web site:

http://www.redbooks.ibm.com/

Once you have completed all server ssh side configurations the follow diagram illustrates the OpenSSH connection for the purpose of accessing the Webmin software. This connection as described in more detail in section 6, illustrates an ssh connection from the client to the server where Webmin resides. The ssh configuration also has set an open port on the client to listen for the Webmin connection. After logging into the server, the user launches their browser pointing to localhost and the local port they configured in the ssh client. For example, if you configured your ssh client to listen on port 10000, then you would point your browser to localhost:10000. The localhost is listening for this and will route the request through the existing ssh connection and to the Webmin application.



Section 3: Downloading ssh Freeware

There are several ssh clients available. Both are easy to download and configure. One such client, called PuTTY, is available at

http://www.chiark.greenend.org.uk/~sgtatham/putty/

Another ssh client is called SecureCRT. Although you can download the product, licenses are required. You can find it at

http://www.vandyke.com/download/securecrt/index.html

The following quote came directly from the Vandyke website:

"In today's security conscious world, it is more important than ever to protect passwords, user accounts, data, and computer systems. SecureCRT provides security for remote access, file transfer, and data tunneling by combining the open Secure Shell protocol with rock-solid terminal emulation".

This point hit's upon the end result of using ssh encryption techniques or tunneling to ensure data is not passed in clear-text. While the Webmin product does support the use of SSL encryption, I have chosen to disable SSL since the ssh tunneling handles all of the security concerns surrounding user ids, passwords, data, and computer systems. In addition if you use Webmin's SSL with ssh and port forwarding, you are essentially doing double encryption checking which will slow down Webmin's response time.

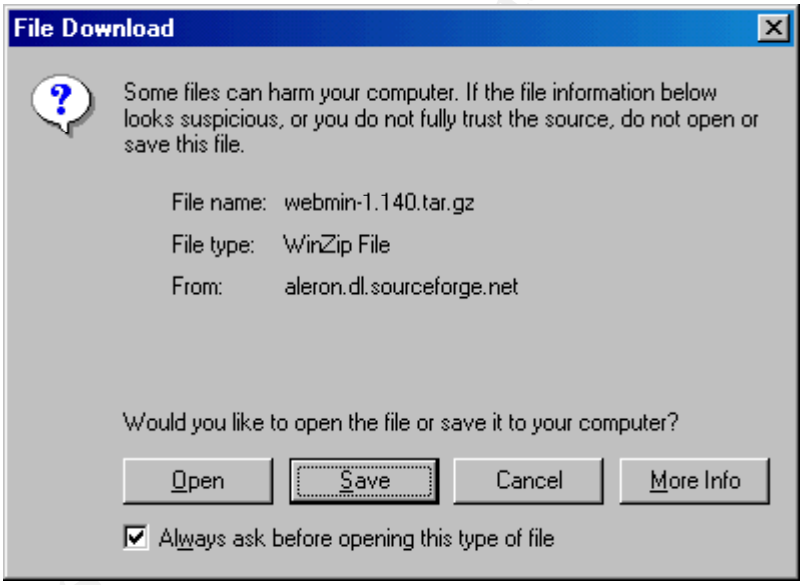
Now that we have discussed both economical and labor cost concerns, as well as using system supported tcp wrappers (In an AIX environment), downloading ssh Free Ware, and the proper configuration for ssh on the server, we are now ready to turn our attention towards the downloading and final securing of Webmin.

Section 4: Downloading Webmin

Webmin can be downloaded, installed and configured -- with several users access rights completed -- within 20 to 30 minutes. The following depicts the instructions on how to download. Portions of this came directly from the Webmin site (http://www.webmin.com/).

How to Download and Install Webmin

1. Go to **http://webmin.com**
2. Select Downloading and Installing from the Webmin main menu.
3. Double click the version of Webmin to download (i.e. webmin-1.140.tar.gz)
4. Next, choose the save option from the download screen and save the file on your PC's hard drive.



   a.
5. At this point you must decide where you want to install Webmin. My choice for this exercise was to use my own home directory. This decision was based on making it easier for me to locate all the Webmin files, instead of traversing the server file directory structure. Installing Webmin this way is not practical or safe when actually installing on any machine that is connected to any network.
6. Before proceeding with the installation you must ensure that you are in the directory of choice. Since I choose my home directory the following command was entered:
   a. cd /home/UserID
   b. mkdir var
   c. cd var
7. Transfer the file to the server and directory where Webmin will be

installed, using either SFTP or WinSCP3. (WinSCP3 can be found at http://winscp.sourceforge.net/eng/download.php)

8. The next step is to unzip and decompress the downloaded file. This was done on Redhat 9.0 using gunzip. The format of the unzip utility is:
    a. gunzip –cd / webmin-1.140.tar.gz | tar –xvf-
    b. The –cd option tells gunzip to write standard output in decompressed format.
    c. webmin-1.140.tar.gz – This is the path to the file copied from the initial download.
    d. –xvf- Tells the tar utility to (x – extract), (v – verbose), and (f – use the tar file).
9. The result of the gunzip created webmin-1.140 in /home/UserID/var. Since I don't want to remember this long name I then linked the directory.
10. Linking is simply a way to create an alias name. I linked webmin-1.140 to the name Webmin. This was done with the following command.
    a. ln –s webmin-1.140 webmin. The –s option makes a symbolic link instead of a hard link.
11. You are now ready to execute the installation of Webmin. Change to your directory where the unzipped and decompressed files reside. In this case the linked directory is webmin.
12. Execute the Perl script called setup.sh. Note: This must be run as root. If you are not root and can't use sudo, then the script will fail. The installation allows you to take directory/file defaults or pick where you want to install the application. As mentioned above, I used my home directory tree to help when looking for parts of the application I installed.
    a. Note: Before choosing the port you will want to check /etc/services. Once you decide on a port, update /etc/services with either a new entry or change the name of the existing entry to reflect "webmin". I initially did not do this, so the service showed up on my port but running under the name of Amanda.
13. Upon completion of the setup.sh Perl script, you will have a full-blown webmin application available. Be sure to remember the administrator username and password you picked during the installation. Also, create a clone administrator account. This will be used in case you locked yourself out, or set a configuration that only the administrator can fix. Better to be safe than sorry.

Section 5: Securing Webmin with tcp wrappers

Once Webmin is loaded and functioning you have opened free access to whatever port you used during the installation. Many companies incorporate security/server standards stating that all services that can be tcp wrapped must be. Since Webmin is running the tcp protocol, tcp wrapping is viable. The intent of tcp wrapping is to secure access to the service and port. Keep in mind when using this approach that it isn't flexible enough if you have a large network, have users accessing Webmin from home either via dial-up, or VPN, or for administrators who may travel to different locations. The wrapping process within AIX depends on IP addresses to secure and finalize the wrapping. This can present issues unless you want to open up all IP's within your network to allow the flexibility to handle the above-mentioned scenarios. For more information visit http://www.itworld.com/AppDev/1076/UIR000630tcp/

The following quote is from http://www.itworld.com/AppDev/1076/UIR000630tcp/

"TCP wrappers are intended to provide wrapper daemons that can be installed without any changes to existing software. Most TCP/IP applications depend on the client/server model -- i.e., when a client requests a connection, a server process is started on the host. TCP wrappers work by interposing an additional layer, or *wrapper,* between client and server. In the basic service, the wrapper simply logs the name of the client host and requested service, then hands this information over to the real daemon; it neither exchanges information with the client or server nor imposes overhead on the actual conversation between the two. Optional features may be enabled, including access control, client-user name lookups, and additional protection against hostname spoofing."

Before starting the Webmin service you should ensure the TCP wrapping is accomplished.

Note: A service can only be tcp wrapped if it is running the tcp protocol. UDP protocols cannot be wrapped.

If the software is already installed you will need to identify the Webmin service name and where the binaries are. If you are installing Webmin then this information is known as you specify file locations during the implementation. To find the Webmin service name you can query your system for services listening. In AIX you can use the command lsof -i | grep LISTEN. You must add an entry in the AIX file /etc/inetd.conf, and it should look something like this:

```
webmin  stream  tcp nowait  root    /usr/local/sbin/tcpd
/usr/local/webmin/miniserv.pl /usr/local/etc/webmin/miniserv.conf
```

An associated entry will be put in /etc/hosts.allow and should look something like this:

miniserv.pl: 127.0.0.1

This accomplishes tcp wrapping to listen for only local connections. Webmin is accessed via your web browser. Once you have tcp wrapped the connection your browser entry would contain localhost:Port (Note: Port equals wherever you choose during the download process.

The issue with this tcp wrapping scenario is no one trying to access the Webmin application from external connection will be allowed entry. This can be a huge issue. The following diagram illustrates ssh tunneling which further secures the application.

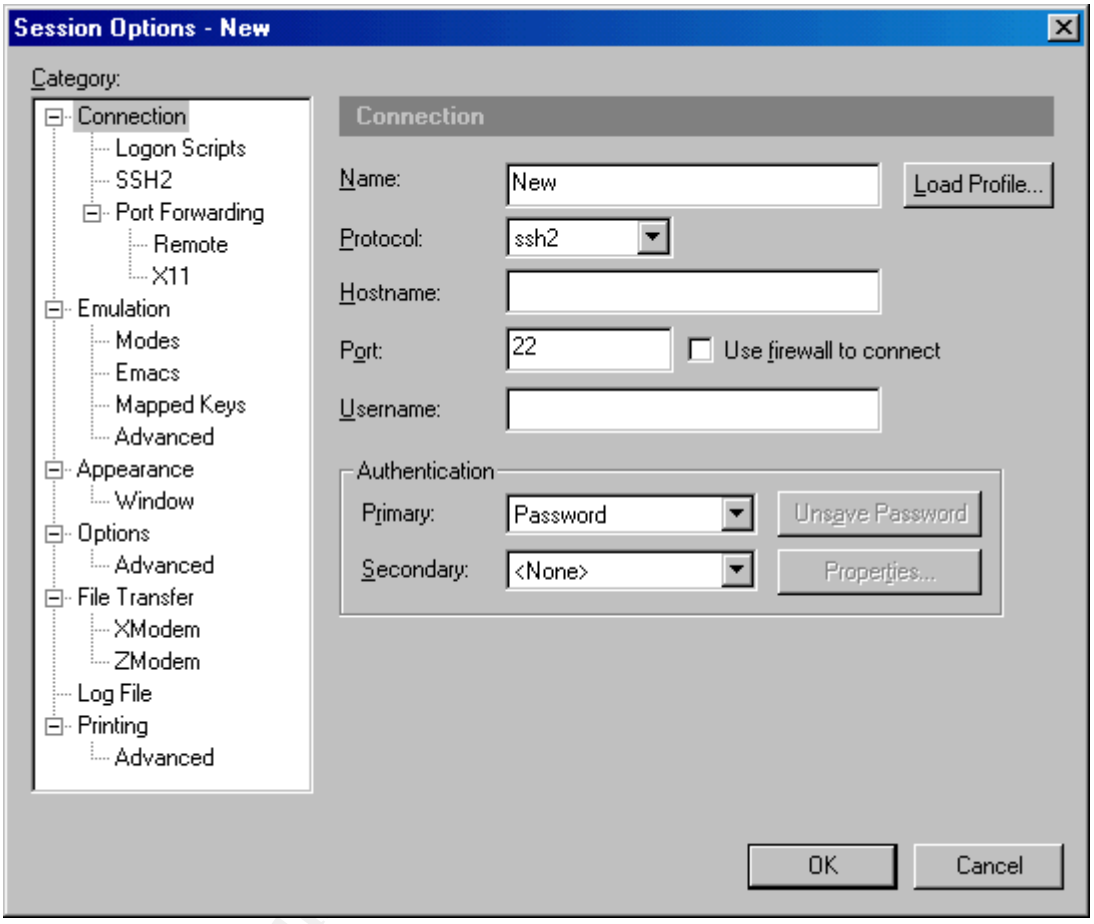
When ssh port forwarding is configured on your client, you will need to select an available port on your localhost and specify the server DNS or ip address (See **Part 4 for screen prints and details)**. This is to establish the local listening option for ssh, which in turn will be used by your browser to forward your request to the Webmin server. On your PC press Start=>Run=>cmd=> netstat –p tcp. Any port not shown can be used.

1. Logon on to the server with your ssh client that has been configured with port forwarding.
    a. This establishes the secure tunnel from your PC to the server.
    b. This also sets your PC's selected port to an active LISTENING mode.
2. Launch your browser to establish the Webmin connection.
    a. After the http:// line, enter localhost:port. The port is what you chose when you configured you ssh client. This is the port listening on your localhost.
    b. Once you have entered this information, your localhost will answer the request since you have activated the port forwarding options as stated above.
    c. Your localhost will then send your request to the server localhost as specified in your configuration. This configuration as explained below in this paper is the server name and port where Webmin is listening.
3. Terminating the ssh connection will kill your Webmin session. Your ssh session can be terminated in several ways such as logging off the server or timing out. I recommend a 20 minute time out option for inactive ssh

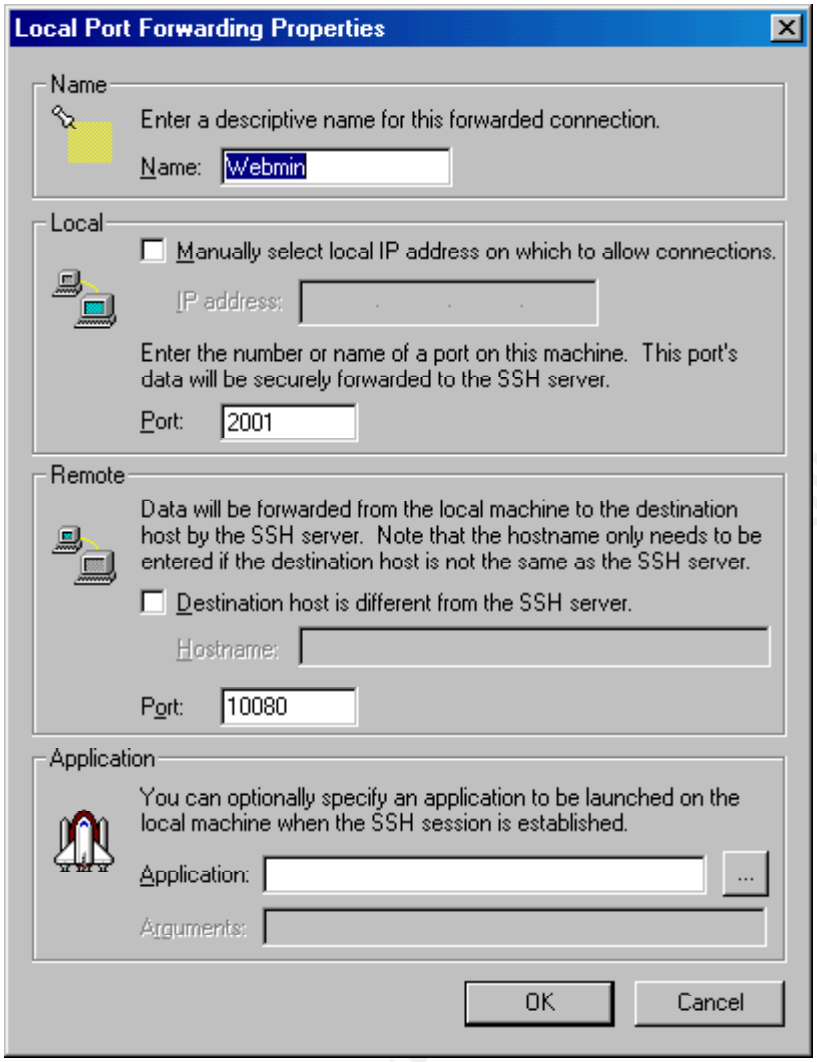
sessions for security purposes.

Section 6: Further securing Webmin with Port Forwarding ssh Clients

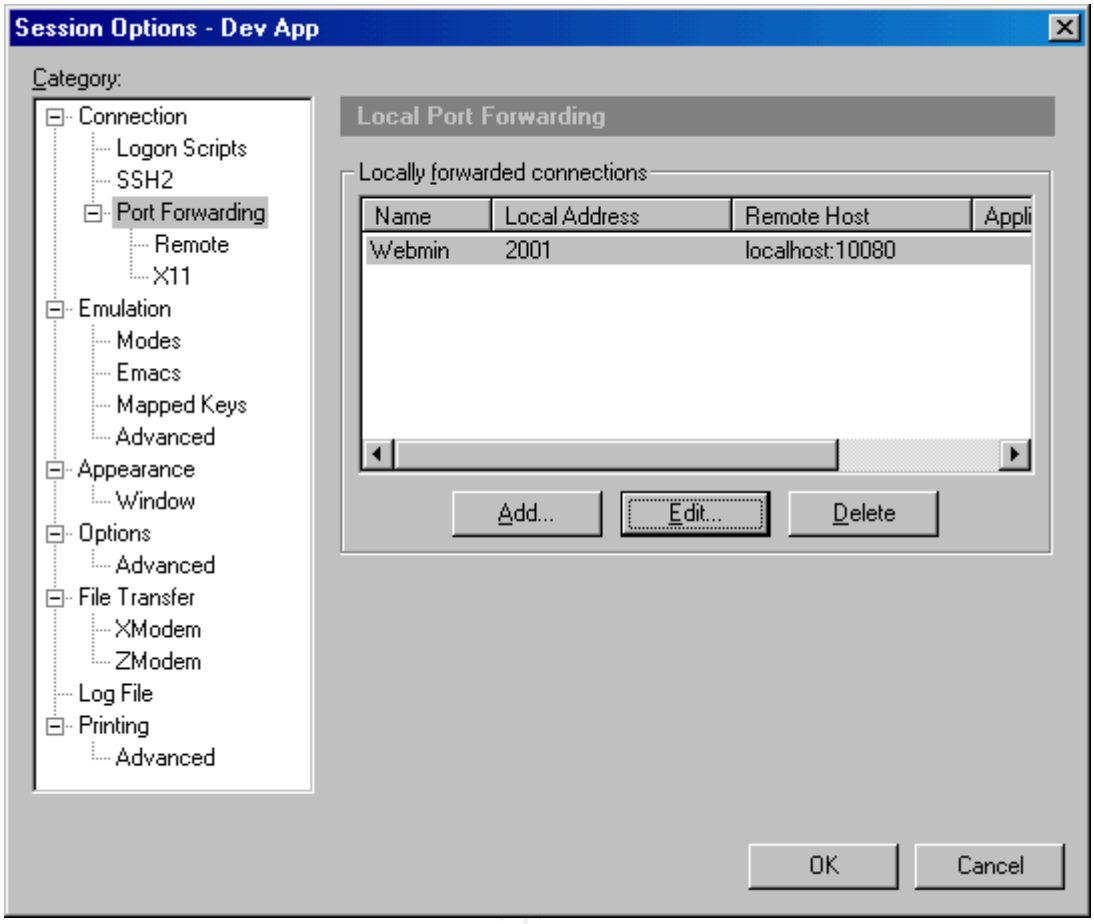
After installing SecureCRT begin to setup your sessions. Here is an example for initial setup. Begin by launching your SecureCRT client.



Click on Port Forwarding and then the add button. Complete the screen as follows:



Click OK.

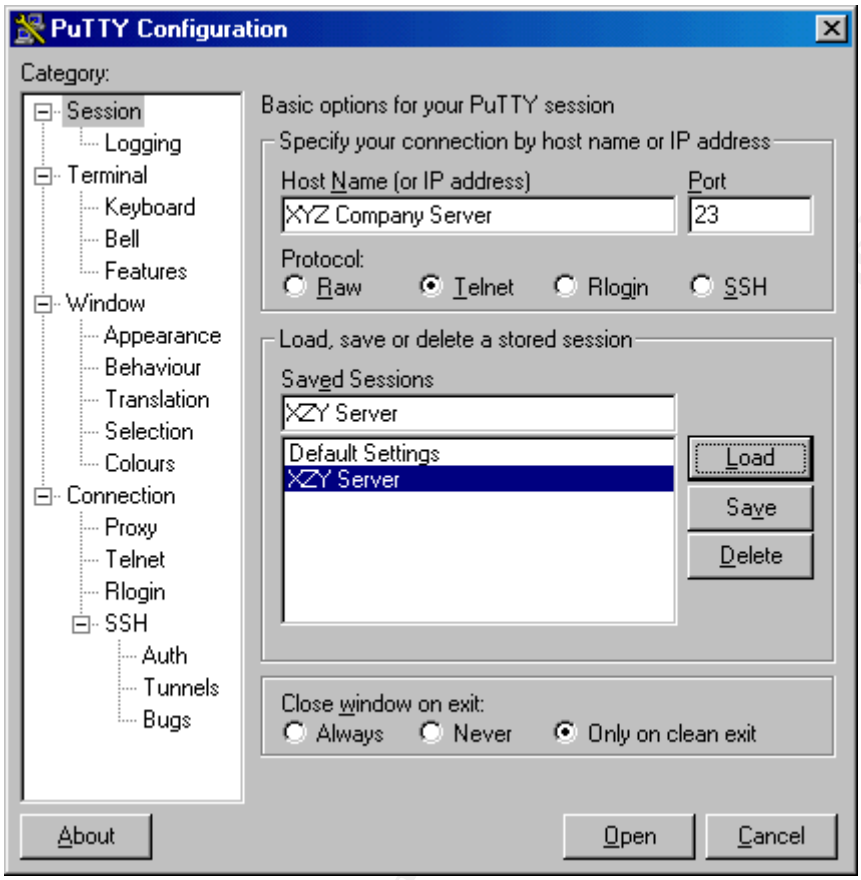


You are now ready to begin the Webmin login process. Launch SecureCRT and connect to the server. After you have logged in and are at the unix command prompt you can minimize the window. Now, launch Internet Explorer. Since Webmin has been secured by the ssh tunnel you have setup by connecting to the server as detailed above, you now need to go through the tunnel and port as follows:

http://localhost:2001/

Notice that you no longer need https. This is because the ssh tunnel is now doing the work that SSL did in the past.

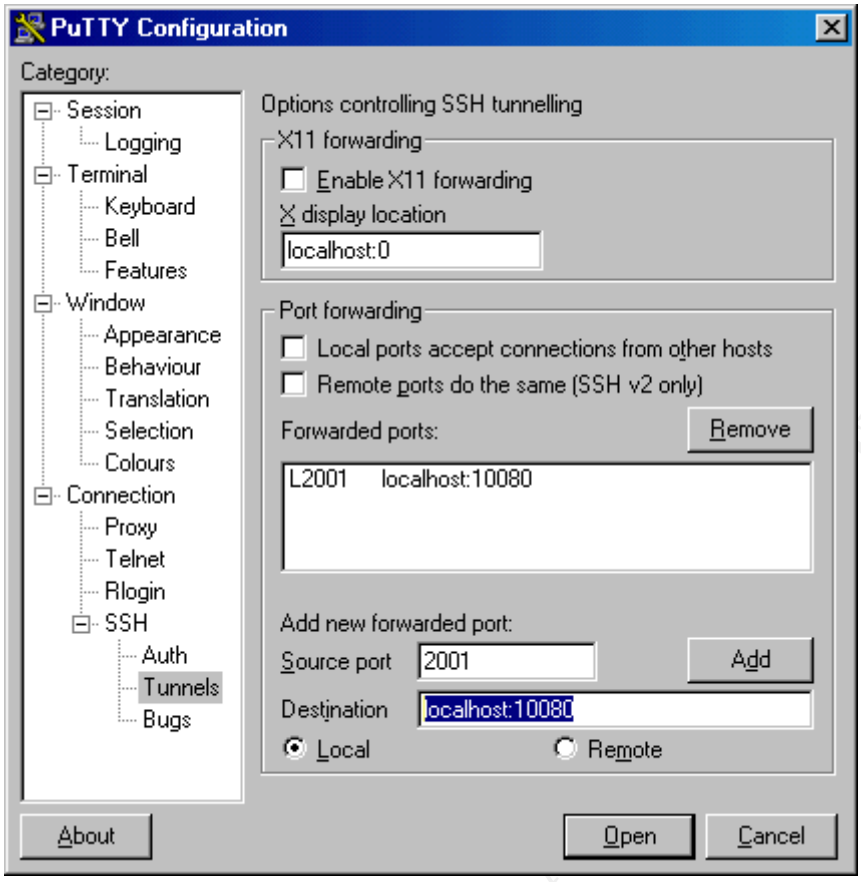
The following example denotes the setup for the PuTTY client.



Click Tunnels listed under SSH and enter the following configuration:

You are now ready to begin the Webmin login process. Launch PuTTY and connect to the server. After you have logged in and are at the unix command prompt you can minimize the window. Now, launch Internet Explorer. Since Webmin has been secured by the ssh tunnel you have setup by connecting to the server as detailed above, you now need to go through the tunnel and port as follows:

http://localhost:2001/



The information is listed above under the Add new forwarded port section. After you complete this hit the Add button and you will see the information as listed above under Forwarded ports:

Bibliography

The book of Webmin or: How I learned to stop worrying and love unix by: Joe Cooper - http://www.swelltech.com/support/webminguide/

Download and Documentation for PuTTY Client.
http://www.chiark.greenend.org.uk/~sgtatham/putty/

Downloading and Documentation for SecureCRT.
http://www.vandyke.com/product/securecrt/

TCP Wrapping.
http://www.itworld.com/AppDev/1076/UIR000630tcp/

How is Port Forwarding Configured?
http://kbserver.netgear.com/kb_web_files/n101145.asp

Evi Nemeth, Garth Snyder, Scott Seebass, Trent R. Hein. Unix System Administration Handbook . New Jersey: Prentice Hall PTR, 1995

The IBM Redbook, *Managing AIX Server Farms.*
http://www.redbooks.ibm.com/

PuTTy.
http://www.chiark.greenend.org.uk/~sgtatham/putty/

SecureCRT.
http://www.vandyke.com/download/securecrt/index.html

Last Updated: May 30th, 2017

Upcoming Training

SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event